

10-5-00

10/03/00



10/03/00  
09/679138  
PTO  
U.S.  
Jc825



**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Only for new nonprovisional applications under 37 CFR 1.53(b))**

Docket No. : 39815/JEJ/X2  
Inventor(s) : Rex Hill, Bryan Dietz and John Bailey  
Title : TUPLE-BASED LOOKUP SCHEME FOR PACKET SWITCHING  
NODE

Express Mail Label No. : EL483174348US

**ADDRESS TO:** Assistant Commissioner for Patents  
Box Patent Application  
Washington, D.C. 20231

Date: October 3, 2000

1.   X   **FEE TRANSMITTAL FORM** *(Submit an original, and a duplicate for fee processing).*

2. **IF A CONTINUING APPLICATION**

       This application is a        of patent application No. .

Prior application information: Examiner ; Group Art Unit:

       This application claims priority pursuant to 35 U.S.C. §119(e) and 37 CFR §1.78(a)(4),  
to provisional Application No. .

3. **APPLICATION COMPRISED OF**

**Specification**

  17   Specification, claims and Abstract (total pages)

**Drawings**

   6   Sheets of drawing(s) (FIGS. 1 to 5)

**Declaration and Power of Attorney**

       Newly executed

  X   Unexecuted declaration

       Copy from a prior application (37 CFR 1.63(d))(for continuation and divisional)

4.        **Microfiche Computer Program** *(Appendix)*

5.        **Nucleotide and/or Amino Acid Sequence Submission** *(if applicable, all necessary)*

       Computer Readable Copy

       Paper Copy (identical to computer copy)

       Statement verifying identity of above copies

6. **ALSO ENCLOSED ARE**

       Preliminary Amendment

       A Petition for Extension of Time for the parent application and the required fee are  
enclosed as separate papers

       Small Entity Statement(s)

       Statement filed in parent application, status still proper and desired

---

**UTILITY PATENT APPLICATION TRANSMITTAL**  
**(Only for new nonprovisional applications under 37 CFR 1.53(b))**

---

Docket No.: 39815/JEJ/X2

---

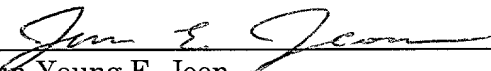
- ☐ Copy of Statement filed in provisional application, status still proper and desired
- ☐ An Assignment of the invention with the Recordation Cover Sheet and the recordation fee are enclosed as separate papers
- ☐ This application is owned by pursuant to an Assignment recorded at Reel , Frame
- ☐ Information Disclosure Statement (IDS)/PTO-1449
- ☐ Copies of IDS Citations
- ☐ Certified copy of Priority Document(s) (*if foreign priority is claimed*)
- ☐ English Translation Document (*if applicable*)
- ☒ Return Receipt Postcard (MPEP 503) (should be specifically itemized).
- ☐ Other

**7. CORRESPONDENCE ADDRESS**

**CHRISTIE, PARKER & HALE, LLP, P.O. BOX 7068, PASADENA, CA 91109-7068**  
**CUSTOMER NUMBER: 23363**

Respectfully submitted,

CHRISTIE, PARKER & HALE, LLP

By   
Jun-Young E. Jeon  
Reg. No. 43,693  
626/795-9900

JEJ/eaj

10-5-00

A

**FEE TRANSMITTAL  
UTILITY PATENT APPLICATION**

**DATE: October 3, 2000**

Docket No. : 39815/JEJ/X2  
Inventor(s) : Rex Hill, Bryan Dietz and John Bailey  
Title : TUPLE-BASED LOOKUP SCHEME FOR PACKET SWITCHING NODE

**FEE DETERMINATION**

CLAIMS AS FILED					
	NUMBER FILED	NUMBER EXTRA	SMALL ENTITY RATE	LARGE ENTITY RATE	FEE
TOTAL CLAIMS	30 - 20	10	x \$9.00	10 x \$18.00	\$180.00
INDEPENDENT CLAIMS	9 - 3	6	x \$40.00	6 x \$80.00	\$480.00
MULTIPLE-DEPENDENT CLAIMS FEE			\$135.00	\$270.00	
BASIC FEE			\$355.00	\$710.00	\$710.00
TOTAL FILING FEE					\$1,370.00
List Independent Claims: 1, 5, 8, 12, 14, 18, 22, 25, 29					

**METHOD OF PAYMENT**

- ☒ Payment Enclosed: Check for \$1,370.00
- ☒ The Commissioner is hereby authorized to charge any fees under 37 CFR 1.16 and 1.17 which may be required during the **entire pendency** of the application to Deposit Account No. 03-1728. Please show our docket number with any charge or credit to our Deposit Account. **A duplicate copy of this sheet is enclosed.**

Respectfully submitted,

CHRISTIE, PARKER & HALE, LLP

By Jan-Young E. Jeon  
Jan-Young E. Jeon  
Reg. No. 43,693  
626/795-9900

JEJ/eaj  
EAJ PAS277221.1\*-10/3/00 4.08 PM

**TUPLE-BASED LOOKUP SCHEME FOR PACKET SWITCHING NODE**

## BACKGROUND OF THE INVENTION

The present invention relates to packet processing and,  
5 more particularly, to tuple-based packet lookup schemes.

Many packet switching nodes classify packets into flows  
in order to facilitate packet processing. Flows are often  
represented by tuples consisting of fields from the packet  
(source address, destination address, etc.) and properties  
10 associated with the packet (ingress port, quality of  
service, etc.). These tuples typically include an ordered  
string of bits representing the various flow properties  
forming the tuple.

In a conventional tuple-based packet processing  
15 operation, the tuple is applied to locate an entry within a  
flow information database having two halves--a key half,  
which matches the tuple, and a result half, which contains a  
payload used for processing packets within a flow defined by  
the tuple. Particularly, when the node receives a packet,  
20 it searches the database to find an entry with a key half  
that matches the tuple from the packet. When such an entry  
is found, the corresponding result half is retrieved and  
used to modify the packet, enqueue the packet for quality of  
service, and/or forward the packet out on one of the node's  
25 ports. This search-and-retrieve operation is commonly  
referred to as a lookup scheme.

One problem commonly encountered in configuring tuple-  
based lookup schemes is key size limitations. For example,  
a node's flow information database may only be able to  
30 accommodate keys containing 80 bits or fewer.  
Unfortunately, this maximum key size may be insufficient for  
a multi-property classification scheme that requires tuples  
having a larger number of bits. Another problem encountered

in tuple-based lookup schemes is how to accomplish efficient "subnetting", i.e. how to effectuate a lookup scheme that provides common processing to a group of distinct flows having some common flow properties in an efficient manner.

5

#### SUMMARY OF THE INVENTION

In one embodiment, the present invention overcomes key size limitations of flow information databases through the implementation of a lookup scheme in which a tuple  
10 representing a plurality of flow properties is parsed into a plurality of subtuples for application in recursive lookups. In a first lookup stage, a first subtuple including a first subset of bits from the tuple is applied to the flow information database and returns a result including a  
15 nickname having a smaller bit count than the first subtuple. In a second lookup stage, a second subtuple including a second subset of bits from the tuple and the nickname are combined and applied to the flow information database. The lookup stages continue until a result indicates that no  
20 recursion is required. The final lookup result includes flow information applicable to one or more of modifying, enqueueing or forwarding the packet.

In another embodiment, the invention supports a truncated lookup capability enabling common processing  
25 across a group of distinct flows having common flow properties. Such common processing may be achieved by returning as part of a result in response to a non-terminal subtuple an indicator specifying that no recursion is required. These and other aspects of the present invention  
30 may be better understood by reference to the following detailed description taken in conjunction with the accompanying drawings briefly described below.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a network environment including a packet switching node;

5        Figure 2 illustrates a representative one of the switching interfaces operative within the packet switching node of Figure 1;

10        Figure 3A illustrates the flow information database operative within the switching interface of Figure 2 undergoing an exemplary IP-based lookup;

Figure 3B illustrates the flow information database operative within the switching interface of Figure 2 undergoing an exemplary truncated IP-based lookup; and

15        Figure 3C illustrates the flow information database operating within the switching interface of Figure 2 undergoing an exemplary MAC-based lookup;

Figure 4 illustrates an alternative flow information database including hashing logic; and

20        Figure 5 is a flow diagram describing a generic tuple-based lookup scheme in accordance with a preferred embodiment of the invention.

## DETAILED DESCRIPTION

25        In Figure 1, network environment including a packet switching node 100 is shown. Node 100 includes a plurality of switching interfaces 120, 121, 122 interconnected to respective groups of LANs 110, 111, 112 and interconnected to each other over data paths 140, 141, 142 via switching backplane 130 and over control paths 150, 151. Switching  
30        interfaces 120, 121, 122 forward packets to and from their respective groups of LANs 110, 111, 112 in accordance with one or more operative communication protocols, such as, for

example, media access control (MAC) bridging and Internet Protocol (IP) routing.

Turning to Figure 2, a representative one of switching interfaces 120, 121, 122, which is designated switching interface 200, is shown in greater detail. Interface 200 includes access controller 210 coupled between LANs and switching engine 220. Controller 210 receives inbound packets off LANs, performs flow-independent physical and MAC layer operations on the inbound packets and transmits the inbound packets to engine 220 for flow-dependent processing. Controller 210 also receives outbound packets from engine 220, performs physical and MAC layer operations on the outbound packets and transmits the packets on LANs. Engine 220 is preferably coupled to many elements for facilitating flow-dependent processing, including flow information database (FIDB) 230 in which lookups are performed.

Particularly, engine 220 receives inbound packets, classifies the packets, generates tuples from the packets in accordance with the classifications, parses selected ones of the tuples into subtuples, applies tuples and subtuples to flow information database (FIDB) 230, accepts results from FIDB 230 returned in accordance with the applied tuples and subtuples, modifies the packets in accordance with flow information from results and transmits the modified packets on switching backplane 130. Engine 220 also receives packets modified by other ones of interfaces 120, 121, 122 from backplane 130, subjects selected ones of the packets to egress processing and transmits selected ones of the packets to controller 210 for forwarding on LANs. Engine 220 may be implemented in well known non-programmable logic, programmable logic or a combination thereof.

Turning now to Figure 3A, FIDB 230 is shown in greater detail undergoing an exemplary IP-based lookup. FIDB 230

includes key half 310 and result half 320. Result half 320 is further divisible into payload portion 321 and recursion indicator portion 322. FIDB 230 includes a first entry consisting of a key portion <ipda> in key half 310, a  
5 payload <nickname> in corresponding payload portion 321 and a recursion indicator <yes> in recursion portion 322. FIDB 230 includes a second entry consisting of a key portion <ipsa, port, qos, nickname> in key half 310, a payload <flowinfo> in corresponding payload portion 321 and a  
10 recursion indicator <no> in recursion portion 322.

In accordance with the IP-based lookup, switching engine 220 receives an inbound packet having an IP destination address <ipda>, an IP source address <ipsa> and a quality of service <qos> on an ingress port <port>,  
15 classifies the packet for IP routing and generates a tuple in the form <ipda, ipsa, port, qos> specified for IP routing. Engine 220 parses the tuple into a first subtuple <ipda> and a second subtuple <ipsa, port, qos> for performing an IP-based lookup.

The first subtuple <ipda> is applied to FIDB 230 and returns a corresponding first payload <nickname> and first recursion indicator <yes> to engine 220, wherein the first payload <nickname> has a smaller bit count than the first subtuple <ipda>. Since the first recursion indicator <yes>  
25 indicates that recursion is required, engine 220 combines first payload <nickname> with the second subtuple <ipsa, port, qos>, applies the combined data to FIDB 230 and returns a corresponding second payload <flowinfo> and second recursion indicator <no> to engine 220. Since the second  
30 recursion indicator <no> indicates that recursion is not required, engine 220 applies the second payload <flowinfo> in processing the packet. The second payload <flowinfo> may include flow information directly applicable in packet



processing, such as, for example, modifying, enqueueing, and forwarding the packet, or may include an index applicable to another database to return flow information directly applicable in packet processing.

5       It will be appreciated that by resolving the first subtuple <ipda> to a first payload <nickname> having a smaller bit count than the first subtuple <ipda> and applying the first payload <nickname> with a second subtuple <ipsa, port, qos> in a recursive lookup, key size  
10 limitations inherent to FIDB 230 that would prevent a single-stage lookup of the complete tuple <ipda, ipsa, port, qos> may be advantageously overcome. Of course, it is possible within the scope of the invention to segment the tuple <ipda, ipsa, port, qos> into three or more subtuples  
15 and conduct recursive lookups by applying the terminal two or more subtuples in combination with respective ones of two or more nicknames returned from FIDB 230 in connection with respective ones of positive recursion indicators.

Turning now to Figure 3B, FIDB 230 is shown in greater  
20 detail undergoing an exemplary truncated IP-based lookup. In this example of a truncated lookup, a negative recursion indicator is returned in response to a non-terminal subtuple to effectuate common processing of all packets having a particular IP destination address <ipda>. FIDB 230 includes  
25 a first entry consisting of a key portion <ipda> in key half 310, a payload <flowinfo> in corresponding payload portion 321 and a recursion indicator <no> in recursion portion 322.

In accordance with the truncated IP-based lookup, switching engine 220 receives an inbound packet having an IP  
30 destination address <ipda>, an IP source address <ipsa> and a quality of service <qos> on an ingress port <port>, classifies the packet for IP routing and generates a tuple in the form <ipda, ipsa, port, qos> specified for IP

routing. Engine 220 parses the tuple into a first subtuple <ipda> and a second subtuple <ipsa, port, qos> for performing an IP-based lookup.

The first subtuple <ipda> is applied to FIDB 230 and  
5 returns a corresponding payload <flowinfo> and recursion indicator <no> to engine 220. Since the recursion indicator <no> indicates that recursion is not required, engine 220 applies the payload <flowinfo> in processing the packet. Application of the second subtuple <ipsa, port, qos> to FIDB  
10 230 is thereby preempted. Of course, the truncated lookup capability provided in the invention is not restricted to effectuating common processing of all packets having a common IP destination address, but may be applied to effectuate common processing of all packets having any  
15 common subset of bits within a tuple for which common processing is desired.

Turning now to Figure 3C, FIDB 230 is shown in greater detail undergoing an exemplary MAC-based lookup. FIDB 230 includes a first entry consisting of a key portion <macda>  
20 in key half 310, a payload <flowinfo> in corresponding payload portion 321 and a recursion indicator <no> in recursion portion 322. In accordance with the MAC-based lookup, switching engine 220 receives an inbound packet having a MAC destination address <macda> on an ingress port,  
25 classifies the packet for MAC bridging and generates a tuple in the form <macda> specified for MAC bridging. The tuple <macda> is applied to FIDB 230 and returns a corresponding payload <flowinfo> and recursion indicator <no> to engine 220. Since the recursion indicator <no> indicates that  
30 recursion is not required, engine 220 applies the payload <flowinfo> in processing the packet.

Turning now to Figure 4, an FIDB 400 in accordance with an alternative embodiment of the invention is shown to

include key hashing logic 410. In accordance with an exemplary IP-based lookup conducted in FIDB 400, a switching engine (not shown) receives an inbound packet having an IP destination address <ipda>, an IP source address <ipsa> and a quality of service <qos> on an ingress port <port>, classifies the packet for IP routing and generates a tuple in the form <ipda, ipsa, port, qos> specified for IP routing.

The engine parses the tuple into a first subtuple <ipda> and a second subtuple <ipsa, port, qos> for performing an IP-based lookup. The first subtuple <ipda> is applied to FIDB 400 wherein its bit count is reduced by hashing logic 410 prior to application to lookup table 420. The reduced first subtuple <ipda-hash> is applied as an initial pointer to lookup table 420 and a linked list of entries in lookup table 420 is walked-down using "next pointers" included in the respective entries until an entry is found that includes an exact match for the first subtuple <ipda>.

When an exact match is found, the corresponding first payload <nickname> and first recursion indicator <yes> are returned to the engine. Since the first recursion indicator <yes> indicates that recursion is required, the engine combines first payload <nickname> with the second subtuple <ipsa, port, qos>, applies the combined data to FIDB 400 and the hash-and-lookup process is repeated for the second subtuple <ipsa, port, qos> whereby a corresponding second payload <flowinfo> and second recursion indicator <no> are eventually returned from the engine. Since the second recursion indicator <no> indicates that recursion is not required, the engine applies the second payload <flowinfo> in processing the packet.

Turning finally to Figure 5, a flow diagram describes a generic tuple-based lookup scheme in accordance with a preferred embodiment of the invention. A packet is received (505) and classified (510). A tuple including one or more flow properties is generated in accordance with the classification (515) and is parsed into multiple subtuples if required (520). The tuple or, if the tuple was parsed the first subtuple, is applied to the FIDB (525) and returns a result including a payload. A check is made to determine if the result indicates recursion (530). If the result does not indicate recursion, the returned payload is the flow information (535) and the flow information is applied to process the packet (540). If the result, however, indicates recursion, the payload is a nickname and the nickname is applied with the next subtuple to the FIDB in a recursive lookup (550).

It will be appreciated by those of ordinary skill in the art that the invention can be embodied in other specific forms without departing from the spirit or essential character hereof. The present description is therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims, and all changes that come within the meaning and range of equivalents thereof are intended to be embraced therein.

We claim:

1. A method for determining packet processing data, comprising the steps of:

receiving a packet;

5 forming a plurality of subtuples for the packet from flow properties associated with the packet;

applying one or more of the subtuples as respective inputs to respective one or more of lookups;

and

10 returning packet processing data as an output from at least one of the lookups.

2. The method according to claim 1, further comprising the steps of:

15 returning a nickname as an output from at least one of the lookups; and

applying the nickname as an input to at least one of the lookups.

20 3. The method according to claim 2, wherein the nickname has a lower bit count than at least one of the subtuples.

4. The method according to claim 1, wherein fewer  
25 than all of the plurality of subtuples are applied as the respective inputs to the respective ones of lookups.

5. A method for determining packet processing data, comprising the steps of:

30 receiving a packet;

forming a tuple for the packet including a plurality of flow properties associated with the packet; and applying one or more of portions of the tuple

to respective consecutive one or more of lookups until packet processing data are returned.

6. The method according to claim 5, further comprising the step of:

returning an indicator with the packet processing data to indicate the return of the packet processing data.

7. The method according to claim 6, wherein the indicator is returned prior to applying all portions of the tuple to the lookups.

8. A method for determining packet processing data, comprising the steps of:

inputting a first lookup key including a first portion of a tuple;

determining a nickname in response to the first lookup key, the nickname having a lower bit count than the first lookup key;

outputting the nickname;

inputting a second lookup key including a second portion of the tuple and the nickname; and

outputting packet processing data in response to the second lookup key.

9. The method according to claim 8, wherein the ones of outputting steps further include outputting respective ones of recursion indicators sufficient to indicate the need for inputting an additional lookup key.

10. The method according to claim 8, wherein the ones

of outputting steps further include outputting ones of indicators, respectively, sufficient to indicate the absence and presence, respectively, of packet processing data.

5           11. The method according to claim 8, wherein the ones Of outputting steps further include outputting ones of indicators, respectively, sufficient to indicate the presence and absence, respectively, of a nickname.

10           12. A method for determining packet processing data, comprising the steps of:  
            receiving a packet;  
            forming a tuple for the packet including a first  
            subtuple identifying a first flow property associated with  
15     the packet and a second subtuple identifying a second flow  
            property associated with the packet;  
            applying the first subtuple to a database element;  
            and  
            returning data from the database element in  
20     response to the first subtuple to preempt application of the  
            second subtuple to the database element.

25           13. The method according to claim 12, wherein the returned data includes packet processing data.

            14     A switching interface for a data communication  
            switch, comprising:  
            an access controller having a port for receiving a  
            packet; and  
30           a switching engine coupled to the access  
            controller for receiving the packet from the access  
            controller, for determining a tuple for the packet including  
            a plurality of flow properties, for transmitting ones of

portions of the tuple to a database element, and for receiving packet processing data from the database element in response to one of the portions.

5        15. The switching interface according to claim 14, wherein the flow properties include a destination address.

10        16. The switching interface according to claim 15, wherein the flow properties include a source address, a port, and a quality of service.

15        17. The switching interface according to claim 14, Wherein the received packet processing data include a plurality of packet flow information.

18. A switching interface for a data communication switch, comprising:  
         means for receiving a packet;  
         means for forming a plurality of subtuples for the  
20 packet from flow properties associated with the packet;  
         means for applying one or more of the  
subtuples as respective inputs to respective one or more of lookups; and  
         means for returning packet processing data as an  
25 output from at least one of the lookups.

19. The switching interface according to claim 18, Further comprising:  
         means for returning a nickname as an output from  
30 at least one of the lookups; and  
         means for applying the nickname as an input to at least one of the lookups.



20. The switching interface according to claim 19, wherein the nickname has a lower bit count than at least one of the subtuples.

5        21. The switching interface according to claim 18, wherein fewer than all of the plurality of subtuples are applied as the respective inputs to the respective ones of lookups.

10        22. A switching interface for a data communication switch, comprising:  
         means for receiving a packet;  
         means for forming a tuple for the packet including a plurality of flow properties associated with the packet;  
15        and  
         means for applying one or more of portions of the tuple in respective consecutive one or more of lookups until packet processing data are returned.

20        23. The switching interface according to claim 22, further comprising:  
         means for returning an indicator with the packet processing data to indicate the return of the packet processing data.

25        24. The switch according to claim 23, wherein the indicator is returned prior to applying all portions of the tuple to the lookups.

30        25. A switching interface for a data communication switch, comprising:  
         means for inputting a first lookup key including a first portion of a tuple;

means for determining a nickname in response to the first lookup key, the nickname having a lower bit count than the first lookup key;

means for outputting the nickname;

5 means for inputting a second lookup key including a second portion of the tuple and the nickname; and

means for outputting packet processing data in response to the second lookup key.

10 26. The switching interface according to claim 25, further comprising means for outputting respective ones of recursion indicators sufficient to indicate a need for inputting an additional lookup key.

15 27. The switching interface according to claim 25, further comprising means for outputting ones of indicators, respectively, sufficient to indicate the absence and presence, respectively, of packet processing data.

20 28. The switching interface according to claim 25, further comprising means for outputting ones of indicators, respectively, sufficient to indicate the presence and absence, respectively, of a nickname.

25 29. A switching interface for a data communication switch, comprising:

means for receiving a packet;

30 means for forming a tuple for the packet including a first subtuple identifying a first flow property associated with the packet and a second subtuple identifying a second flow property associated with the packet;

means for applying the first subtuple to a database element; and

means for returning data from the database element in response to the first subtuple to preempt application of the second subtuple to the database element.

- 5           30. The switching interface according to claim 29,  
wherein the returned data further includes packet processing  
data.

[illegible]



Figure 1

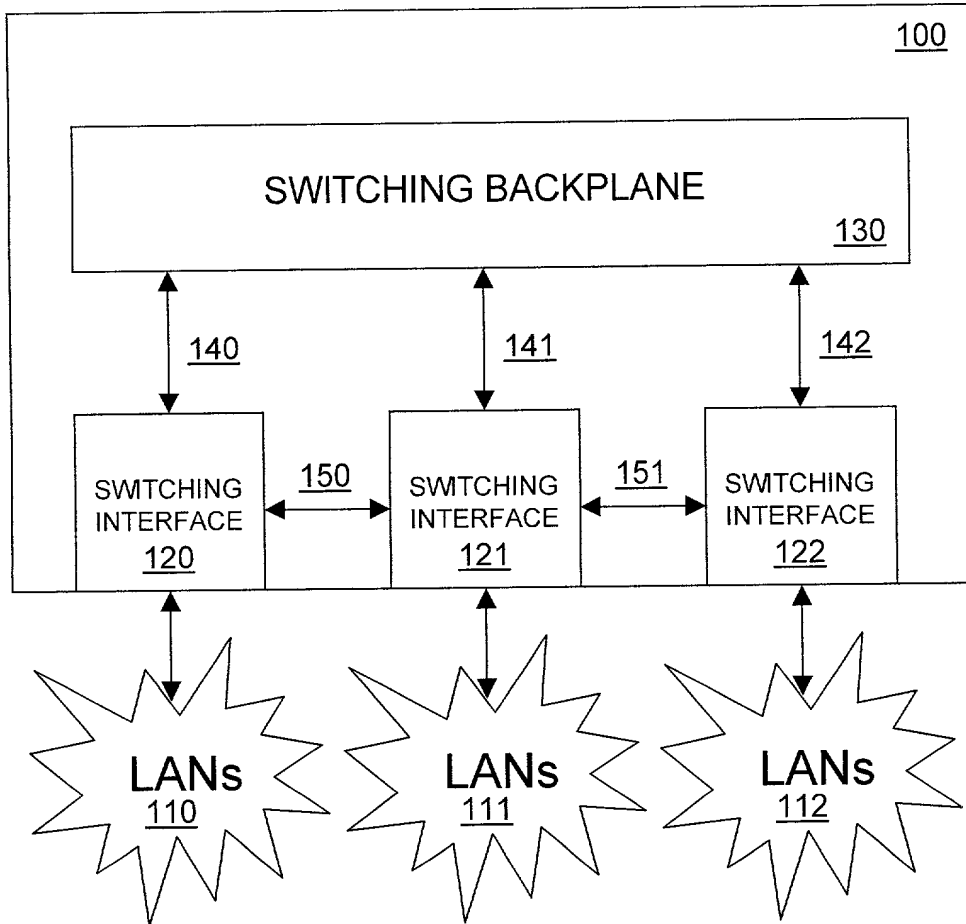
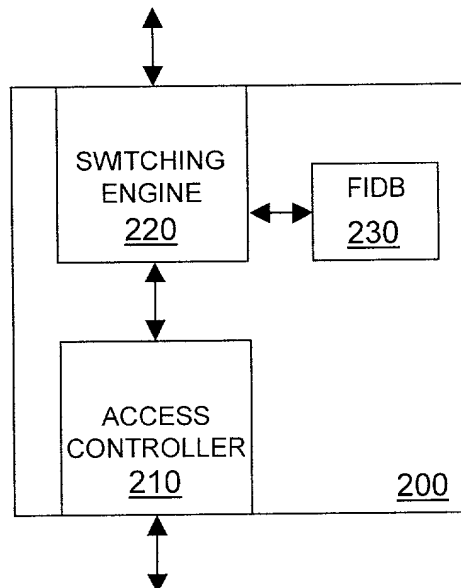


Figure 2



230

Figure 3A

tuple = <ipda, ipsa, port, qos>

KEY <u>310</u>		RESULT <u>320</u>	
		PAYLOAD <u>321</u>	RECURSION <u>322</u>
<u>&lt;ipda&gt;</u>	<u>&lt;ipda&gt;</u>	<u>&lt;nickname&gt;</u>	<u>&lt;yes&gt;</u>
	<u>&lt;ipsa, port, qos, nickname&gt;</u>	<u>&lt;flowinfo&gt;</u>	<u>&lt;no&gt;</u>

230

Figure 3B

tuple = <ipda, ipsa, port, qos>

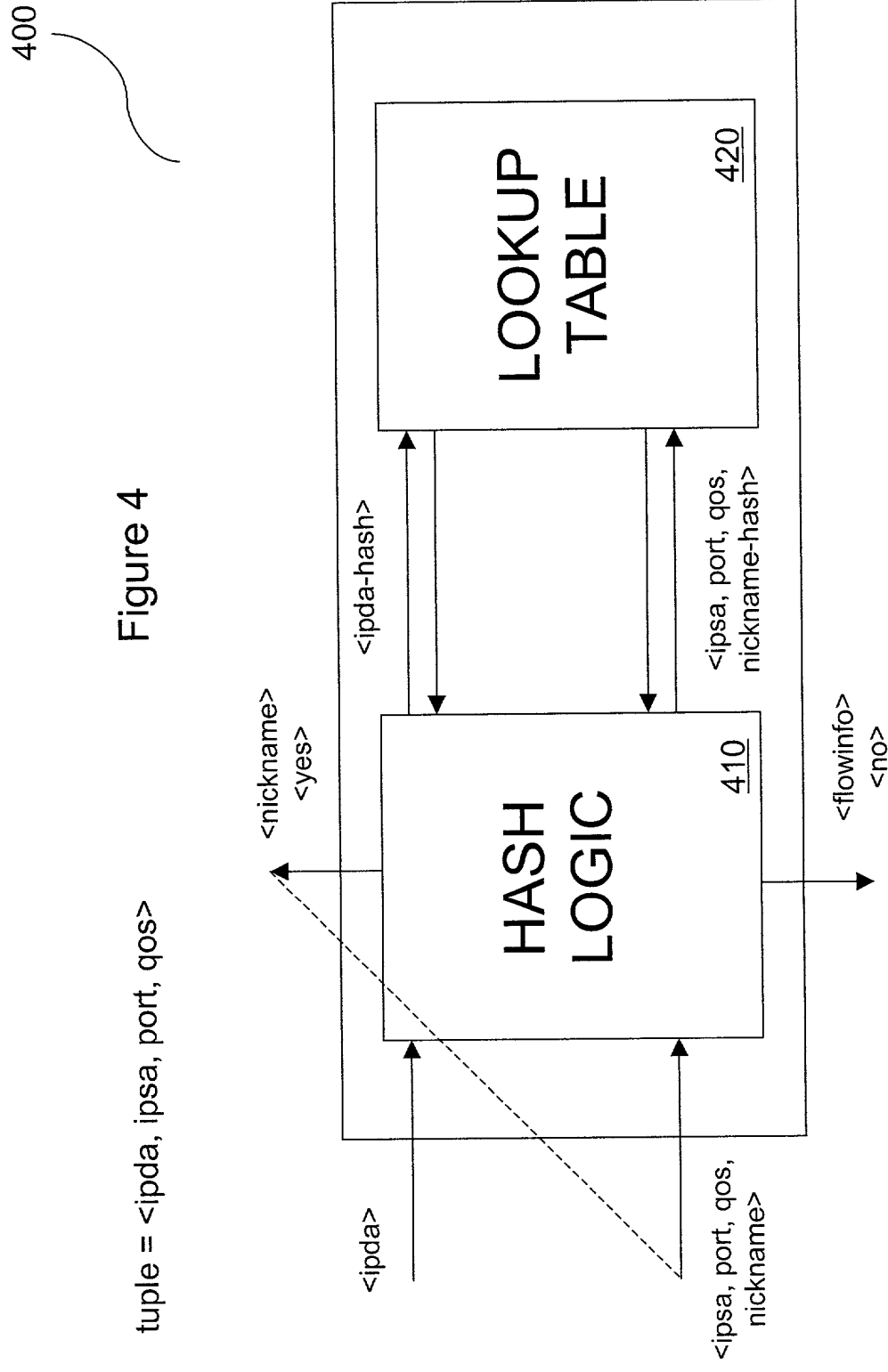
KEY <u>310</u>		RESULT <u>320</u>	
		PAYLOAD <u>321</u>	RECURSION <u>322</u>
<u>&lt;ipda&gt;</u>		<flowinfo>	<no>

<ipda>

<flowinfo>







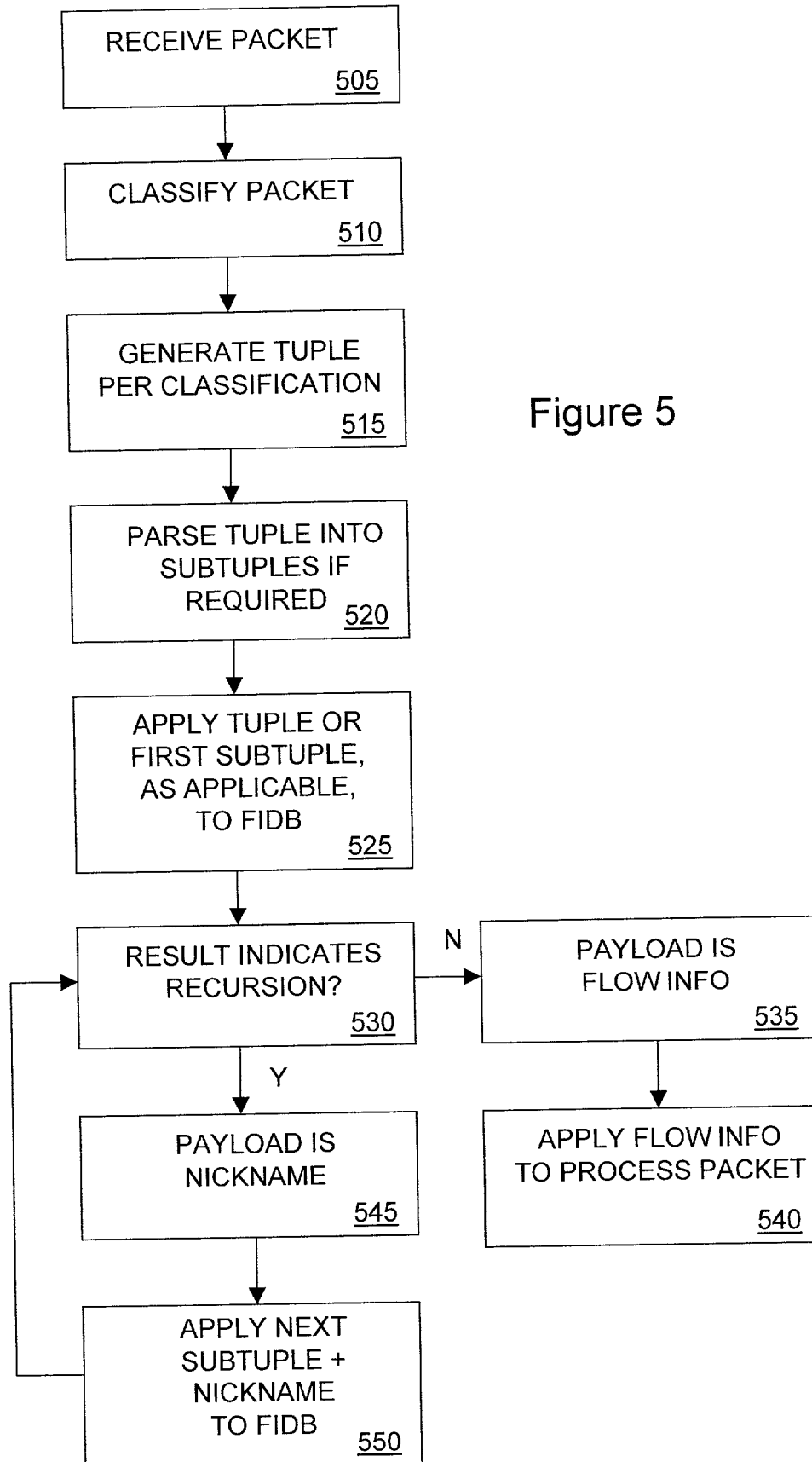


Figure 5

**DECLARATION AND POWER OF ATTORNEY  
FOR PATENT APPLICATIONS**

PATENT

Docket No. : 39815/JEJ/X2

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled TUPLE-BASED LOOKUP SCHEME FOR PACKET SWITCHING NODE, the specification of which is attached hereto unless the following is checked:

X was filed on October 3, 2000 as United States Application Number or PCT International Application Number \_\_\_\_\_ (to be filled in later when known) and was amended on \_\_\_\_ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or § 365(b) of the foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

<u>Application Number</u>	<u>Country</u>	<u>Filing Date (day/month/year)</u>	<u>Priority Claimed</u>
---------------------------	----------------	-------------------------------------	-------------------------

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below.

<u>Application Number</u>	<u>Filing Date</u>
---------------------------	--------------------

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s), or any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

<u>Application Number</u>	<u>Filing Date</u>	<u>Patented/Pending/Abandoned</u>
---------------------------	--------------------	-----------------------------------

**POWER OF ATTORNEY:** I hereby appoint Scot A. Reader (39,002) of Alcatel Internetworking, Inc. and the following attorneys and agents of the law firm CHRISTIE, PARKER & HALE, LLP to prosecute this application and any international application under the Patent Cooperation Treaty based on it and to transact all business in the U.S. Patent and Trademark Office connected with either of them in accordance with instructions from the assignee of the entire interest in this application; or from the first or sole inventor named below in the event the

**DECLARATION AND POWER OF ATTORNEY  
FOR PATENT APPLICATIONS**

**Docket No. 39815/JEJ/X2**

application is not assigned; or from \_\_\_ in the event the power granted herein is for an application filed on behalf of a foreign attorney or agent.

R. W. Johnston	(17,968)	Grant T. Langton	(39,739)	Marc A. Karish	(44,816)
D. Bruce Prout	(20,958)	Constantine Marantidis	(39,759)	John F. O'Rourke	(38,985)
Hayden A. Carney	(22,653)	Daniel R. Kimbell	(34,849)	Richard J. Paciulan	(28,248)
Richard J. Ward, Jr.	(24,187)	Craig A. Gelfound	(41,032)	Josephine E. Chang	(46,083)
Russell R. Palmer, Jr.	(22,994)	Syed A. Hasan	(41,057)	Frank L. Cire	(42,419)
LeRoy T. Rahn	(20,356)	Kathleen M. Olster	(42,052)	Harold E. Wurst	(22,183)
Richard D. Seibel	(22,134)	Daniel M. Cavanagh	(41,661)	Robert A. Green	(28,301)
Walter G. Maxwell	(25,355)	Molly A. Holman	(40,022)	Derrick W. Reed	(40,138)
William P. Christie	(29,371)	Lucinda G. Auciello	(42,270)	John W. Peck	(44,284)
David A. Dillard	(30,831)	Norman E. Carte	(30,455)	Stephen D. Burbach	(40,285)
Thomas J. Daly	(32,213)	Joel A. Kauth	(41,886)	David B. Sandelands, Jr.	(46,023)
Vincent G. Gioia	(19,959)	Patrick Y. Ikehara	(42,681)	Heidi L. Eisenhut	(46,812)
Edward R. Schwartz	(31,135)	Mark Garscia	(31,953)	Nicholas J. Pauley	(44,999)
John D. Carpenter	(34,133)	Gary J. Nelson	(44,257)	Mark J. Marcelli	(36,593)
David A. Plumley	(37,208)	Raymond R. Tabandeh	(43,945)	Donald Bollella	(35,451)
Wesley W. Monroe	(39,778)	Cynthia A. Bonner	(44,548)		
Gregory S. Lampert	(35,581)	Jun-Young E. Jeon	(43,693)		

The authority under this Power of Attorney of each person named above of the law firm shall automatically terminate and be revoked upon such person ceasing to be a member or associate of or of counsel to that law firm.

**DIRECT TELEPHONE CALLS TO : Jun-Young E. Jeon, 626/795-9900**

**SEND CORRESPONDENCE TO : CHRISTIE, PARKER & HALE, LLP  
P.O. Box 7068, Pasadena, CA 91109-7068**

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first joint inventor Rex Hill	Inventor's signature	Date
Residence and Post Office Address San Diego, California		Citizenship US

Full name of second joint inventor Bryan Dietz	Inventor's signature	Date
Residence and Post Office Address Lake Forest, California		Citizenship US

---

---

**DECLARATION AND POWER OF ATTORNEY  
FOR PATENT APPLICATIONS**

---

---

**Docket No. 39815/JEJ/X2**

---

---

Full name of third joint inventor John Bailey	Inventor's signature	Date
Residence and Post Office Address Agoura Hills, California		Citizenship US